

Climb, Fly, Stack: Design of Tangible and Gesture-based Interfaces for Natural and Efficient Interaction

Alexandre Audinot^{1,*}

Emeric Goga^{1,2,*}

Vincent Goupil^{1,2,*}

Carl-Johan Jorgensen^{1,2,*}

Adrien Reuzeau^{1,*}

Ferran Argelaguet^{1,*}

¹Univ Rennes, Inria, CNRS, IRISA, ²SATT Ouest Valorization

ABSTRACT

This paper describes three novel 3D interaction metaphors conceived to fulfill the three tasks proposed in the current edition of the IEEE VR 3DUI Contest. We propose the *VladdeR*, a tangible interface for Virtual ladder climbing, the *FPDrone*, a First Person Drone control flying interface, and the *Dice Cup*, a tangible interface for virtual object stacking. All three interactions take advantage of body proprioception and previous knowledge of real life interactions without the need of complex interaction mechanics: climbing a tangible ladder through arm and leg motions, control a drone like a child flies an imaginary plane by extending your arms or stacking objects as you will grab and stack dice with a dice cup.

Index Terms: H.5.2 [User Interfaces]: User Interfaces—Graphical user interfaces (GUI);

1 INTRODUCTION

This year the IEEE VR 3DUI contest required the design of interaction techniques to perform three different tasks: ladder climbing, drone flying and object stacking. This paper describes our solution, which focused on the design of tangible and gesture-based input mechanisms to resemble real-life interactions. We believe that the described techniques can inspire further research on 3DUI.

2 CLIMBING A VIRTUAL LADDER

The virtual climbing task required the user to climb a virtual ladder in which the user could control independently his/her feet and hands. Our goal was to provide a good matching between the user's proprioceptive information and the sensory data in order to achieve higher levels of presence [2]. We wanted the user to feel that he/she was actually climbing a ladder. Di Loreto and al. [1] have already shown the effectiveness of using a physical ladder to enhance presence. However, they used a standard ladder which has a predefined number of rungs, which does not allow to climb taller virtual ladders.

VladdeR. In our system, we wanted the user to be able to climb any virtual ladder, regardless of the number of rungs. With this intention, we designed a dynamic tangible interface for climbing virtual ladders: the *VladdeR* (see Figure 1). Our first physical prototype consisted of a 2.5 meters high wooden structure stabilized with lateral supports and equipped with four movable rungs, one for each limb. Each rung is allowed to move vertically along two vertical rails with a span equal to the distance between two consecutive virtual rungs. An elastic mechanism brings the rung up whenever the user releases his hold of it, and a metallic contact detects when it reaches its highest position.

Climbing task control. In the user's resting position, all four rungs are brought down and the virtual hands and feet of the user are attached to their respective virtual rungs. When the user releases a



Figure 1: Left, the *VladdeR*, a tangible interface for Virtual ladder climbing. Right-up, the hands rungs. Right-down, the feet rungs.

rung, it is brought back up by an elastic mechanism and a contact is detected and translated into a keyboard press using a *Makey Makey*¹. Then, the corresponding virtual hand or foot is free to move and the user can grab/step on the rung again. Every time that the user descends a rung with any of their hands the virtual camera is raised, giving the impression that the user is hauling himself up. In addition to their head, users' hands were tracked using HTC Vive trackers.

3 FLYING A VIRTUAL DRONE

The drone task required the user to control a drone in order to flight through a pre-defined route (go through a set of ring-shaped objects). This section describes the main components of *FPDrone*, a novel navigation metaphor for First Person Drone control. Our goal was to provide a gesture-based interface which enabled the coupling of all the degrees of freedom of the drone at once, taking advantage of body proprioception.

Input Mapping. The designed navigation interface can be seen as a head-steering flying interface [4] in which hand motions determine roll and speed. Head tracking was provided by the HTC Vive and hands motions by two HTC Vive trackers. The drone was driven by computing (A) a target drone up-vector (defined by the head yaw and the signed difference between the left and right hand height (see Algorithm 1), (B) a support plane (π) in which the drone forward vector should lay (defined by the user's viewing direction and the global y-axis) and (C) a target thrust value (defined by the distance between both hands). Regarding the user's view, in order to reduce simulation sickness, it did not reflect the current drone orientation, the view direction was directly controlled by head motions.

Stabilization. In order to ease the drone control, the system automatically compensates for inertia and external forces (e.g. gravity

*e-mail: name.surname@irisa.fr

¹<https://makeymakey.com>

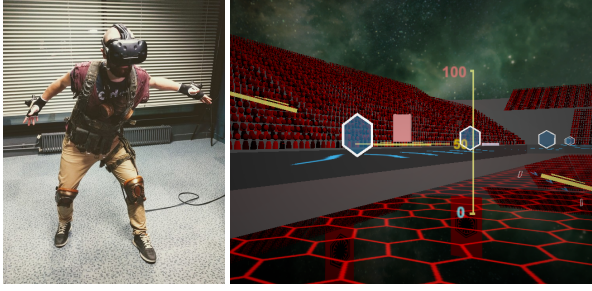


Figure 2: Illustration of the FPDrone interface. Left, hand tracking was achieved using two HTC Vive trackers. Right, the GUI of FPDrone provided information regarding pitch (vertical slider), roll (horizontal lines) and thrust (vertical line).

input : x : head yaw; y : signed hand height difference
output : Target up vector in head space

```

 $x \leftarrow \text{clamp01}(x, \text{minYaw}, \text{maxYaw});$ 
 $y \leftarrow \text{clamp01}(y, \text{minDiff}, \text{maxDiff});$ 
 $\mathbf{v} \leftarrow (x, y) / \|(x, y)\|;$ 
if  $\|\mathbf{v}\| < 0.01$  then return  $(0, 0, 1);$ 
 $\phi \leftarrow \angle((1, 0), \mathbf{v});$ 
 $\theta \leftarrow (1 - \|\mathbf{v}\|)\pi/2;$ 
return  $(\cos(\theta)\sin(\phi), \cos(\theta)\cos(\phi), \sin(\theta))$ 

```

Algorithm 1: Computation of the target up-vector. A spherical notation is used to describe the hemisphere determined by the positive z-axis (view direction). The head yaw $[-35..90]$ deg contributed to changes in the elevation (pitch), while the difference on height $[-70..70]$ cm contributed to changes in lateral motion (roll).

and drag). The compensation system updates the target up-vector and the target thrust value. For example, when the user required the drone to stop, the system will compute an up-vector opposing the drone motion reducing the braking time. In addition, once the drone is stopped, the system will compute a thrust which will negate gravity. Automatic stabilization greatly improved the maneuvering of the drone; users had a lot of difficulties to compensate inertial forces manually. If not enabled, it resulted on the feeling of losing control of the drone, which can potentially increase the risk of motion sickness [3].

Control. Once the new target values were computed, the system computed two rotations. The first rotation aims to align the drone to the target up-vector, and the second rotation aims to align the forward vector with π . Both rotations are concatenated and a yaw-pitch-roll command is sent to the drone. The closed-loop control ensured smoothness and responsiveness.

Graphical User Interface. The interface (see Figure 2) provided information regarding the current state of the drone. (1) A vertical slider placed in the center of the view provided information regarding the pitch control (head pitch). (2) A second vertical slider, placed in the center right provided information regarding the current thrust (hand distance). (3) Two vertical lines displaying the current roll value were also displayed at the sides. The most important feedback, elevation and speed were kept at the center to ensure their visibility.

4 STACKING VIRTUAL OBJECTS

The stacking task required the user to be able to stack a set of objects. In order to perform this task, we propose the **Dice Cup** metaphor (see Figure 3). The user controls a virtual dice cup which enables to grab and stack virtual objects as if they were dice. With the **Dice Cup**, users can perform two operations: grab virtual objects and the release of all virtual objects previously grabbed.

Object Grab. Grabbing an object is achieved by covering the



Figure 3: Illustration of the Dice Cup metaphor. From left to right, tracked tangible cup, grab operation, release confirmation after shaking the cup and lift of the cup revealing the stacked objects.

desired virtual object with the virtual cup. Upward red arrows are displayed on the cup to display a grabbing confirmation. The user can grab as many objects as desired.

Object Release. In order to release the grabbed objects, the user has to shake the virtual cup. Once the release is confirmed (blue downward arrows on the cup), the user rests the cup on the virtual table (red downward arrow displayed) and then, he lifts it to release all previously grabbed objects. The objects are released in the inverse order as grabbed. In addition, objects are not structured in a perfect stack they fell on each other following their physical properties following the y-axis of the cup.

Furthermore, in order to enable the precise control of the virtual cup and increase immersion, we provided a tangible object (plastic cup) which matched the virtual dice cup. The tangible cup was tracked using a HTC Vive tracker. Furthermore, we added small wooden objects inside the plastic cup to provide sound and tactile feedback when shaking the cup.

5 RESULTS AND CONCLUSION

Informal testing among the colleagues in the lab has shown the all three interfaces provides usable interfaces for the completion of the three tasks. Moreover, users with no experience are able to complete the tasks without major problems. Yet, we observed two main aspects that require additional analysis/works. First, the Vladder cannot match perfectly the preconception of the user with his virtual situation. Indeed, when the virtual hand/feet are on different rungs, the user's hands/feet are at the same level. Second, the control of the drone at maximum thrust required moderate training. The drone is controlled by the laws of physics, thus the user must learn the limits of the drone and which maneuvers are possible.

In conclusion, although further evaluations are needed, we believe that the proposed techniques provides natural and intuitive mechanisms to perform the three proposed tasks in the IEEE VR 3DUI Contest enabling an efficient control with a limited amount of training.

REFERENCES

- [1] C. Di Loreto, J.-R. Chardonnet, J. Ryard, and A. Rousseau. Simulation de travail en hauteur en réalité mixte : étude de la présence et de l'anxiété. In *Actes des douzièmes Journées de l'Association Française de Réalité Virtuelle*, pages 99–104.
- [2] M. Slater, M. Usoh, and A. Steed. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):201–219, 1995.
- [3] K. M. Stanney and P. Hash. Locus of user-initiated control in virtual environments: Influences on cybersickness. *Presence: Teleoperators and Virtual Environments*, 7(5):447–459, 1998.
- [4] M. Slater, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr. Walking, walking-in-place, flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 359–364, 1999.